1 Misc. Programming Questions

1. Write a method, *reverse*, which takes one argument, a **String**, and returns a String which is the reverse of the first. For example, *reverse("hello")* should return "olleh".
2. Write a program that reads in an integer, n, from the user, and then prints out the first n letters of the alphabet. For example, for $n = 6$, the output should be *abcdef*.
   (a) Are there any upper or lower bounds on what the value of n can be? If so, what are they?
   (b) Come up with some test cases to show that your program works without error.
3. Given an integer array, *arr*, write a program to compute the following:
   (a) The average of the array
   (b) The sum of the array
   (c) The mode of the array (mode is the number that occurs the most number of times).
4. Write a method, *strToArr*, which takes one argument, a **String**, and returns a character array that represents the String. For example, *strToArr("hi")* should return $\{'h','i'\}$
5. Write a program that, given a number, n, creates an array where each index, i, holds the i^{th} digit of n. IE, given 12345, we would create the array: {1,2,3,4,5}
6. Write a program that, given two integer arrays, a and b, creates an array c which is a concatenated with b. IE, $a = \{1, 2, 3\}$, and $b = \{4, 5, 6\}$, $\therefore c = \{1, 2, 3, 4, 5, 6\}$.
7. Write a program that reverses a String, *str*, and prints the result to the console.
8. Write a program that finds the number of times a String, *target*, occurs in a String, *str*. For example, if *str* equals "hello hehe hie", and *target* equals "he", you should output 3.
9. Write a program that, given an array of integers, *ints*, computes:

$$\sum_{i=0}^{n} (ints[i])^2 - ints[i] + 2$$

10. Write a method called *upperCount* that, given a String *str* as argument, returns the number of characters in *str* are upper case letters. For example, *upperCount("HeLlo");* should return 2.
11. Write a method body for the following method, which will take two integer arrays as input and find their dot product. That is, given arrays a and b, you are finding: $\sum_i (a[i] * b[i])$:

```
public static int dotproduct(int[] a, int[] b){




}
```

2 Object Oriented Programming

For the questions on the next page, we will use the following class definition:

```
class Dinosaur{
   private int height;   // The height of a dinosaur in feet
   private int weight; // The weight of a dinosaur in pounds
   private String color; // The color of the dinosaur

   public Dinosaur(int height, int weight, String color){
      this.height = height;
      this.weight = weight;
      this.color = color;
   }

   public Dinosaur(String color){
      this.color = color;
      this.height = 0;
      this.weight = 0;
   }

   public void eat(int mass){
      if(isNegative(mass)) return;
      this.weight += mass;
   }

   public void grow(int amount){
      if(isNegative(amount)) return;
      this.height += amount;
   }

   private boolean isNegative(int value){
      if(value < 0) return true;
      return false;
   }

   public String getColor(){
      return this.color;
   }

   public int getHeight(){
      return this.height;
   }

   public int getWeight(){
      return this.weight;
   }

   public void equals(Object object){
      return this == object;
   }

}
```

12. Suppose you want to create a Dinosaur object, which is red, 100 feet tall, and weighs 300 pounds. You want to store this object in a variable called *dino*. Write the statement that would accomplish this:

---

13. Suppose you wrote the following lines, written after your declaration in the previous question. For each line, indicate whether there will be an error, or if there is no error, what the result of the line will be:
    (a) dino.getWeight();
    (b) dino.setWeight(50);
    (c) Dinosaur.isNegative(-50);
    (d) dino.height += 50;
    (e) dino.eat(50);
    (f) dino.eat(-50);
    (g) dino.equals("Hello world");

14. This question asks several questions about the .equals method in the Dinosaur class. The equals method, in this class, is meant to return true if the other object is a Dinosaur object, isn't null, and if the height, weight, and colors of both dinosaurs are identical. That is, if one Dinosaur is 5 feet tall, weighs 50 pounds, and has a color of "purple", it should only return true if the other Object is a Dinosaur that is 5 feet tall, weighs 50 pounds, and has a color of "purple".
    (a) Does the .equals method in the Dinosaur class currently work correctly as it is supposed to?
    (b) What happens if you call *dino.equals("Hello")* in the current implementation?
    (c) How do you check if *other* is a Dinosaur object or another type of Object?
        (a) Write a statement to check if *other* is a String object:
        (b) Write a statement to check if *other* is a Dinosaur object:
    (c) Now, write a new version of equals for the Dinosaur class that behaves correctly:

```
public boolean equals(Object other){




}
```

15. Fill out the body of the below method, which should return how many of the Dinosaur objects in the array argument *arr* are equal to the Dinosaur object argument *dino*:

```
public static int numEquals(Dinosaur dino, Dinosaur[] arr){




}
```

3 Arrays, Strings, and Errors

16. For each of the following, assume we have declared the following:

```
int[] arr = {1, 2, 3, 18};
int[] arr2 = new int[8];
String s = "Hello World";
```

Identify what the result will be, or write *Compile-time* or *Run-time* if there is a Compile-time or Run-time error:

(a) arr[0];
(b) arr[arr2.length - 5];
(c) s.charAt(arr.length);
(d) s.charAt(arr[3]);
(e) s.charAt(arr[5]);
(f) s.charAt(arr[1]);
(g) s.length;
(h) s.charAt(s.indexOf('H'));
(i) s.charAt(s.indexOf('Q'));
(j) s.indexOf('Q');
(k) arr2[0];
(l) arr2 = arr;
(m) arr[0] = (int) s.charAt(6);
(n) arr[0] = s.charAt(6);
(o) int j = 6.0;
(p) double d = 7;
(q) s += 'q';
(r) (s + "Goodbye!").length();
(s) s.equals("Hello " + "World");

17. Consider this code snippet:

```
int[] array = {4, 5, 6, 7, 8};
int mystery = 0;
for(int i = 0; i < array.length; i ++){
    mystery += array[i];
}
int mystery2 = array.length;
System.out.println((mystery * 1.0) / mystery2);
```

What is the output from the following code snippet?

How many total times does the loop run?

In words, what does this snippet do?

18. Now consider this code snippet:

```
int alakazam = 0;
for(int i = 0; i < 8; i ++){
    for(int j = 0; j < i; j ++){
        alakazam += j;
    }
}
```

What is the output from the following code snippet?

How many total times does the inner loop run?

In words, what does this snippet do?

4 Movie Project

In the first Movie project, we asked you to create a class which modelled a Movie. The following questions will ask you to write some methods involving that project.

19. Write the body to the following method, which will return an array containing the Movie(s) with the highest rating. That is, if there are 5 Movie objects, two of which with a 3 star rating, and 3 of which with a 1 star rating, you should return an array of size 2 containing just the ones that achieved 3 stars. If there are 5 different movies all with different rankings, you should return an array of size 1 which contains the movie with the highest rank:
Hint: You should first find the highest rating of all of the movies, then compute how many of the Movie objects have that rating, and then create an array of sufficient size and add each movie. In effect, you might want to go through the array three times.

```
public static Movie[] highestRankedMovies(Movie[] movies){




}
```

20. Write the body of the following method, which will return the average rating of all of the Movie objects in the array:

```
public static double averageRating(Movie[] movies){




}
```

21. Write the body of the following method, which will return which Movie is the *smallest*. You may assume all of the Movie objects in the array are distinct (ie: none are equal or have the same name).
Hint: You should use the compareTo method in the Movie class.

```
public static Movie smallestMovie(Movie[] movies){




}
```